auditing itself can in part be traced back to the U.S. government. In the 1970's, the U.S. Department of Defense ("DOD") funded an extensive research effort to provide computer system security for the processing of classified information. As part of this program, the DOD created a policy for implementing certain auditing functions to satisfy certain security goals, including "to allow the review of patterns of access" and "to allow the discovery of both insider and outside attempts to bypass protection mechanisms, and to assure that "attempts to bypass the protection will be recorded and discovered…".[58]

93. James P. Anderson is often cited as the first person to propose that computer auditing mechanisms could be used by computer security personnel to track internal and external penetrations and misfeasance.[59] After Anderson's early work in 1980, several other early intrusion detection systems were developed which focused upon the analysis of audit trail information. Some of the initial analysis of audit trails was actually performed by hand, with security personnel manually reviewing audit trails to look for problems and patterns. Eventually, given the labor-intensive nature of such work, the industry began to move towards more automated systems for reviewing audit trials to detect misuse of computer systems.

94. The 1980s also saw a shift in computing generally away from more mainframe and centralized computing to networks. As networks proliferated, network monitoring tools were developed to look for network faults and communication errors. Eventually the intrusion detection community followed this trend, and intrusion detection systems began focusing upon network traffic and network sources for attack. This shift towards incorporating network-based sources of data for intrusion detection occurred

patents-in-suit also state that the inventions were made with government support provided by DARPA.

[58] *See* R. Bace, INTRUSION DETECTION at 11.

[59] *See* J. Anderson, *Computer Security Threat Monitoring and Surveillance*, Washington, PA, James P. Anderson Co., 1980.

much earlier than the alleged inventions in the patents-in-suit, however. By 1990, a research group at the University of California, Davis, of which I was an integral part, had already implemented an intrusion detection system using network packets as a data source.

95. The computer industry as a whole has grown enormously in the past 25 years. The field of computer security has also grown drastically in that time. While the intrusion detection community has certainly grown from its initial beginnings in the late 1980s, it has remained a rather small, tight-knit group of people. I attended numerous conferences on intrusion detection in the 1990s in particular, and there was certainly a core group of people in attendance who were well-known to each other.

96. Cooperation and collaboration among practitioners in the intrusion detection field has often occurred. For example, the Common Intrusion Detection Framework, or CIDF, was a joint effort by many of the researchers funded on a Defense Advanced Research Projects Agency (DARPA) Information Technology Office (ITO) program led by Teresa Lunt. The DARPA Program Manager, Teresa Lunt, and the DARPA researchers recognized that a lot of researchers were attempting to solve similar problems in intrusion detection, so in order to reduce duplicative work and to foster interoperability between the technology components and systems being developed by the researchers, they decided to jointly develop CIDF. For example, during the DARPA Principal Investigator's (PI) meeting in Savannah in February of 1997, the CIDF effort's proposed vision was: "To develop standards so that all ITO funded Intrusion Detection Systems can demonstrably inter-operate." The kick-off slides further state: "Six projects are, at least in part, developing an 'architecture or system for other components,'"

including Boeing, SRI, UC Davis, Purdue, GE, and North Carolina (MCNC and NCSU).[60]

97. Over the next several months the CIDF members developed a series of proposals for a common network protocol, message syntax and semantics, and APIs to support interoperability between their components. The CIDF members actively worked together, communicating regularly over email (a mailing list was established), meeting at computer security related conferences, and meeting during a summer 1997 DARPA PI meeting held at SRI. Over the summer and early fall of 1997 several groups submitted proposals to the mailing list covering different aspects of CIDF. For example, the EMERALD/JiNao team submitted a proposal in July presenting a protocol for event components, analysis components, and countermeasure components to communicate with each other.[61]

98. In addition to Teresa Lunt's DARPA ITO program, other DOD-related efforts proclaimed they would support CIDF as well. For example, the DARPA Information Assurance program led by Sami Saydjari, of which I was part, also planned to use CIDF to support communication between their intrusion detection components.

99. Eventually the DARPA researchers attempted to broaden the effort and make CIDF an endorsed standard, so they took their proposed design to the Internet Engineering Task Force (IETF). While not endorsing the proposed CIDF solution, IETF did endorse the goal of developing interoperability standards between intrusion detection components and formed the Intrusion Detection Working Group with the goal of

[60] Common Intrusion Detection Framework "APIs" Presentation by Stuart Staniford-Chen 2/26/97 [SYM_P_0071471-SYM_P_0071481].

[61] Email from P. Porras to CIDF, subject "Interface Spec: SRI, MCNC/NCSU," date 22 Jul 1997 [SYM_P_0500624-SYM_P_0500639].

developing RFC standards, the most prominent of which was called the Intrusion Detection Message Exchange Framework (IDMEF).

100. SRI was certainly well-known as an important group in the intrusion detection community. Similarly, a group of people including myself from the Computer Science Laboratory at UC Davis were also well-known as longstanding participants in the intrusion detection community. Both SRI and UC Davis were primarily research-focused. Below I have provided a short overview of the work done by these two research groups. By no means, of course, was research in intrusion detection limited to these two groups. Other important groups doing research in intrusion detection included, for example, Purdue University, Columbia University, MIT's Lincoln Laboratory, UC Santa Barbara, Lawrence Livermore National Laboratory, and Los Alamos National Laboratory.

101. The U.S. government funded a great deal of research in intrusion detection, but also actually used such systems to protect government assets, and I have provided a short history of some of these systems as well. In addition, I have provided a short history of some of the first commercial intrusion detection systems.

1. SRI's history in intrusion detection

102. SRI, in conjunction with various government research efforts, has worked and published in the IDS field for several decades. Much of this published work involves a project developed for and funded primarily by the US Navy and the US Air Force that has undergone three different evolutions over time: IDES, NIDES, and EMERALD. As the inventors themselves have explained:

> Our earlier intrusion-detection efforts in developing IDES (Intrusion Detection Expert System) and later NIDES (Next-Generation Intrusion Detection Expert System) were oriented toward the surveillance of user-session and host-layer activity. This previous focus on session activity within host boundaries is understandable given that the primary input to intrusion-detection tools, audit data, is produced by mechanisms that tend to be locally administered within a

> single host, or domain. However, as the importance of network security has grown, so too has the need to expand intrusion-detection technology to address network infrastructure and services. In our current research effort, EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances), we explore the extension of our intrusion-detection methods to the analysis of network activity.[62]

103. Different authors at SRI, including the named inventors for the patents-in-suit, published extensively on IDES, NIDES, and EMERALD prior to filing the '338 patent.[63]

104. In the late 1980s under a US Navy government contract, SRI began working on IDES – an audit-based system to observe computer behavior and learn to recognize "normal" behavior and deviations from expected behavior.[64] IDES used statistical "profiles" of past behavior to describe normal behavior, with the data involved being drawn from audit records. Rare behavior was referred to as "anomalous." IDES eventually also included a rule-based system to detect known security violations.

105. Several other companies and researchers in the late 1980s also developed intrusion detection systems that used statistical methods to find anomalies in audit records. Examples include: (1) Haystack from Tracor Applied Sciences, Inc. / Haystack Laboratories, principally written by Steve Smaha, (2) Multics Intrusion Detection and Alerting System (MIDAS) from the National Computer Security Center (NCSC); (3) Network Audit Director and Intrusion Reporter (NADIR) from Los Alamos National Laboratory and (4) Wisdom and Sense from Los Alamos National Laboratory.[65]

[62] P. Porras and A. Valdes, *Live Traffic Analysis* at 3.
[63] *See* Appendix ---, listing 20 different SRI publications on IDES, NIDES, and EMERALD, all dated more than one year prior to November 9, 1998.
[64] H. Javitz and A. Valdes, *The SRI IDES Statistical Anomaly Detector*, http://www.sdl.sri.com/papers/stats91/, May 1991.
[65] *See* Rebecca Bace, "Intrusion Detection" (Macmillan Tech. Pub. 2000) ("Bace"), at 103-07.

106. Under additional contracts from the US Navy, SRI's "next-generation" of the IDES project (NIDES) extended SRI's work on statistical profiling.[66] IDES and the original NIDES were primarily host-based systems, deriving their information from audit data.[67] However, in the early 1990s researchers were increasingly focusing upon the use of network traffic in IDS, including the government agencies funding SRI's NIDES project. In 1995, SRI published D. Anderson, T. Frivold and A. Valdes, "Next-generation Intrusion Detection Expert System (NIDES) A Summary," May 1995 (*"Network NIDES")*, which advocated for, and explained the ease of, expanding the NIDES system to "Network NIDES." Network NIDES was designed to extend NIDES to cover network monitoring, and specifically described extending NIDES profiles to directly monitor network packets.[68] *Network NIDES* also explained that NIDES could be extended to support hierarchical monitoring via multiple monitors.[69]

107. Under additional government contracts from the US Navy and the US Air Force, SRI incorporated these changes into the successor to NIDES – the EMERALD system. By December 1996 SRI had published a conceptual overview of the EMERALD

[66] R. Jagannathan et al. (including A. Valdes), *System Design Document: Next-Generation Intrusion Detection Expert System (NIDES)*, March 9, 1993 at 55.
[67] T. Lunt, *Detecting Intruders in Computer Systems*, 1993 Conference on Auditing and Computer Technology at 1-2.
[68] *Network NIDES* at 31 ("A network monitoring process could be incorporated into NIDES to read network packets and produce canonical NIDES audit records for analysis."). Note that the decision to "extend" NIDES to include network packet monitoring occurred some four years after U.C. Davis researchers had already published and implemented such an idea for their NSM IDS system. SRI and P. Porras in particular were certainly aware of the NSM project, since it is discussed and cited in *Emerald 1997*. *See Emerald 1997* at 364 and 365 [7].
[69] *Network NIDES at 31* ("In addition, the NIDES architecture could be extended to support multiple cooperative NIDES processes that would each be responsible for a local domain, with a higher-level NIDES process responsible for the network that supports all the local domains.").

system,[70] and by October 1997, SRI had fully disclosed the EMERALD system in the *Emerald 1997* publication. *Emerald 1997* implemented the changes suggested in *Network NIDES* – specifically, monitoring network traffic via monitoring packets, and a system of hierarchical monitors.

108. On November 9, 1998, P. Porras and A. Valdes filed the patent application which matured into the '338 patent. The '203, '212, and '615 patents all stem from the original '338 patent disclosure. These patents relate to the EMERALD system.

109. After the '338 patent filing, SRI continued to work on the EMERALD system. SRI also continued to receive additional government contracts for work on EMERALD.[71] As discussed elsewhere in my report, SRI also filed additional patent applications on EMERALD-related work not disclosed in the patents-in-suit.

2. UC Davis's history in intrusion detection

110. Various researchers at the University of California, Davis have also contributed substantially to the IDS field over several decades. Beginning in the late 1980s, UC Davis developed the Network Security Monitor (NSM) for Lawrence Livermore National Laboratory (LLNL), which is widely credited as being the first IDS that monitored network traffic.[72] LLNL secured funding from the Department of Energy (DOE) and approached UC Davis's Professor Levitt to be the Principal Investigator (PI) on the project. Professor Levitt was well known for his work in computer security, was

[70] P. Porras and P. Neumann, *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances, Conceptual Overview,* http://www.sdl.sri.com/papers/emerald-position1/ (December 18, 1996).
[71] *See* SRI 005524-25, discussing contract F30602-99-C-0149, which SRI has identified as being associated with Emerald. *See* SRI's Second Supplemental Response to Symantec's Interrogatories Nos. 5 and 8 (#8).
[72] R. Bace, INTRUSION DETECTION at 19; SRI 097123-86 at 097167-68, 097178.

previously Director of the SRI Computer Science Laboratory. While Professor Karl Levitt was the PI, I was the primary developer for the NSM project.

111. The UC Davis work on NSM was carried out from 1988 through about 1995, although the primary work was performed from 1988 to 1993. Following 1993 the Air Force took a copy of the NSM and developed it further under the name Automated Security Incident Measurement (ASIM), and along with the Common Intrusion Detection Director (CIDD), continues to develop it to this day. Also around 1993 LLNL took their copy of NSM and further developed it under the name Network Intrusion Detector (NID) for the DOE and provided NID to the Defense Information Systems Agency (DISA) under the name Joint Intrusion Detection System (JIDS).

112. The first major system to integrate host and network-based monitoring was the Distributed Intrusion Detection System (DIDS), introduced in the early 1990s. In addition to integrating network and host information, DIDS also implemented a monitoring hierarchy, and correlated evidence of activity across a network. The first phase of DIDS from 1990-1992 was funded by the United States Air Force, with Lawrence Livermore National Laboratory (LLNL) as the prime contractor and UC Davis and Haystack Laboratories as subcontractors.

113. In 1993 DIDS was handed over to Trident Data Systems (TDS) to develop into a supported product to be deployed across the Air Force. TDS continued to work on DIDS until about 1996, at which point TDS and the Air Force redirected their focus towards Automated Security Incident Measurement (ASIM), along with the Common Intrusion Detection Director (CIDD).

114. UC Davis also conceived and developed the Graph-based Intrusion Detection System (GrIDS). UC Davis began development of GrIDS in 1995 as part of an initial 1993 DARPA contract to develop intrusion detection systems for very large networks. UC Davis continued the GrIDS work under a 1996 contract, and continued to

work on it until about 1999. The GrIDS system was running publicly at UC Davis by approximately early 1997.

3. Government systems' history in intrusion detection

115. The United States government was one of the earliest sponsors and adopters of intrusion detection technologies and systems. For example, the MIDAS intrusion detection system was running on the NSA's Dockmaster computer by the late 1980s.[73] However, wide-spread adoption of intrusion detection systems within the government did not occur until the deployment of the NSM (later called the ASIM sensor) throughout the Air Force in the early to mid-1990s. In addition to the Air Force's ASIM effort, the NSM was used by the Defense Information Systems Agency (DISA), initially as just NSM and then under the name Joint Intrusion Detection System (JIDS), as well as by the Department of Energy in general and Lawrence Livermore National Laboratory in particular under the name Network Intruder Detector (NID).

116. With the availability of commercial network-based intrusion detection systems in the mid to late 1990s, the government started adopting commercial IDSs as well. In particular, NetRanger was extensively evaluated by the DOD and the Air Force bought a number of systems.[74] ISS's RealSecure also became very popular in the government, and I recall one of DARPA's Grand Challenge efforts centering around the analysis of sensor logs from RealSecure. When meeting with government officials in the early 2000s, however, I was finding more and more government sites adopting the open source system called Snort. During an Intelligence Community (IC) Principal Investigators meeting in 2005, the only intrusion detection system I heard people using (besides some of their own research efforts) was Snort.

[73] B. Mukherjee et al., "Network Intrusion Detection," IEEE Network, May/June 1994.
[74] *See* Expert Report of Daniel Teal.

117. It is fair to say that today there is a wide mix of government off the shelf (GOTS), commercial off the shelf (COTS), and open source intrusion detection systems being used within the government, and most of these systems are network-based intrusion detection systems.

4. Commercial systems' history in intrusion detection

118. Haystack Laboratories was one of the first commercial companies developing intrusion detection systems. While beginning their efforts with government contracts (the Haystack intrusion detection system for the Air Force in the late 1980s and DIDS in the early 1990s), they eventually developed their own commercial intrusion detection systems around 1993 with Stalker (a host-base IDS). Stalker was quickly followed by NetStalker (a network-based IDS) and WebStalker (an application-based IDS). In the mid-1990s ISS, which started with a vulnerability scanner, developed a network-based intrusion detection system called RealSecure, and WheelGroup, formed by several people who had worked with the NSM in the Air Force, developed NetStalker. During the late 1990s, in part fueled by the Internet stock bubble, a large number of commercial intrusion detection companies formed. Today, intrusion detection systems are part of the product lines of many companies including Cisco, Juniper, and Symantec.

119. See also the expert report of Frederick Avolio for a history of related network entities such as packet filters and firewalls, and the expert report of Daniel Teal for the history of NetRanger.

X. LACK OF NOVELTY

120. The following sections discuss the features and functionality of each of the main prior art publications and systems that in my opinion demonstrate that some or all of the asserted claims are invalid as anticipated.

A. PRIOR ART PUBLICATIONS AND SYSTEMS

1. NSM

121. The Network Security Monitor (NSM) was a network intrusion detection system first developed in the late 1980s at UC Davis that proliferated to many different organizations and has undergone many different permutations since its inception. Many different publications have described various aspects of the NSM system, as listed in Exhibit C. This report focuses on the NSM as it was described in: L.T. Heberlein, G.V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, D. Wolber, *A Network Security Monitor*, Proc. 1990 Symposium on Research in Security and Privacy, pp. 296-304, May 1990 ("*NSM 1990*").

122. The NSM was designed to protect a network from attack. The target system to protect included "a number of computers (including devices such as file servers, name servers, printers, etc.) and a LAN through which the hosts are inter-connected."[75] The LAN included "the wire, bridges, routers, and gateways."[76] Thus, NSM received packets from a network entity such as a router or a gateway.

123. The NSM project was inspired, to a large extent, by the attacker that Cliff Stoll tracked across the globe – that attacker also crossed DOE networks. Few if any of the penetrated computers generated audit trails that could be processed by intrusion detection systems, but in each case the attacks did generated network packets. The NSM was therefore developed to process packets as the data source. The NSM processed TCP/IP packets on an Ethernet, but *NSM 1990* clearly states that the approach should work on other protocols and network topologies.

[75] *NSM 1990* at 297.

[76] *NSM 1990* at 297.

124. While the NSM used network traffic for its data source, *NSM 1990* clearly states that it uses statistical detection analysis concepts from host-based intrusion detection systems such as SRI's IDES.[77] This was not unusual – statistical analysis techniques, including SRI's statistical profiling techniques, were widely known and had been widely shared with the intrusion detection community. Specifically, the NSM used a combination of anomaly-based statistical detection using profiles and signature-based detection using rules to detect potentially intrusive behavior.

125. The NSM had the following salient features:

Designed to protect hosts (including file servers, name servers, printers, etc.) and a LAN (defined as the wire, bridges, routers, and gateways).

Implemented for the TCP/IP protocol on a CSMA/CD network.

Extendable to other network protocols and network topologies.

Processed network packets.

Modeled several different types of objects including hosts, data paths, and connections.

Monitored network connections and network data volume.

Used both anomaly detection and signature detection to determine if an object was behaving suspiciously.

Responded to detected suspicious activity by reporting objects to a security officer and potentially modifying analysis.

Stated plan to extend the approach to a distributed environment with multiple network monitors exchanging information.

126. The NSM modeled several subjects that shared a hierarchical relationship with each other: source objects, source-destination objects, source-destination-service objects, and connections. These subjects, examples of each, and their relationships are shown in Figure 1 of Exhibit W. The following discussion refers to the number labels in Fig. 1.

[77] *NSM 1990* at 296.

127. The top-level objects are source objects (Src) – hosts that initiate connections. Figure 1 shows two source objects: Venus (1) and Mars, and provides examples of the types of measurements that would be propagated through the NSM. The NSM tracked two measurements for each source object: the number of packets (Pkts) for connections initiated by the host and the number of hosts to which this host tried to connect.

128. The second level objects are source-destination objects (Src-Dst) – these represent pairs of communicating hosts. For example, (2) represents activity in which the host Venus initiated connections to the host Mercury. The NSM tracked two measurements for each source-destination object: the number of packets over connections originating from the source to this destination and the number of different services over which the source communicated to the destination.

129. The third level objects are source-destination-service objects (Src-Dst-Srvc) – pairs of hosts communicating over a specific service such as Telnet, FTP, or Sendmail. For example, (3) represents activity from connections initiated by the host Venus to the host Mercury over the Telnet service. The NSM tracked two measurements for each source-destination-service object: the number of packets over connections originating from the source to this destination over this service and the number of different connections between the pair of hosts and this service.

130. The fourth level objects are the connections themselves (Connections). For example, (4) represents a single connection initiated by Venus to Mercury over the Telnet service. The NSM tracked two measurements for each connections object: the number of packets over the connection and the number of bytes in those packets.[78]

[78] *NSM 1990* mentions only the number of packets, but the NSM system itself tracked both the number of packets and bytes.

131. To determine if a subject was behaving maliciously and a security officer should be alerted, the NSM used three different methods: anomaly detection on the metrics for each subject, signatures on the metrics for each subject, and anomaly detection on the existence of data paths.

132. NSM's anomaly detection used the values from past instances of objects to build a profile of that object's behavior, compare new instances of the object against the profile, and report objects that were behaving anomalously, as shown in Figure 2. NSM's anomaly detection process began by collecting measurements of past historical activity (see top level of Figure 2). For example (1) shows a past instance of a subject that had 121 packets, and (2) shows a past instance of a subject that had 71 packets.

133. NSM took these past instances and built a probability distribution that measured how frequently past values occurred – this was the *profile* for the measurement. (see second level of Figure 2). For example, in Figure 2 the X-axis for the probability distribution represents the number of packets seen for a measurement – on the far left the first bar represents instances which had 0-9 packets, the next one represents instances that had 10-19 packets, and so on (3). The Y-axis represents how frequently that measurement was observed (4). For example, the bar on the far left of the X-axis (0-9 packets) is very small, indicating that very few past measurements had this value (roughly 1%), while the bar for objects with 50-59 packets were the most common, occurring almost 20% of the time. This probability distribution was exponentially aged.

134. The NSM then compared this long-term statistical profile to recent activity. To determine if the difference between the historical and recent activity was anomalous enough to be reported, the NSM security officer set a threshold (5) for the measurement, and any future measurement that fell in a range (or bin) with a frequency below that threshold was reported. This is functionally identical to comparing the

difference between the profiles to the "score threshold" of the patents-in-suit.[79] To implement this approach, the NSM used the concept of a mask (6).

135. A mask (6) is an object that by default, when presented with a value, blocks that value. However, holes (7) can be put into the mask that will allow some values to flow through the mask. Values that flow through the mask are reported. The NSM created an anomaly mask (6) (see third level of Figure 2) and cut holes (7) in the mask corresponding to each bin in the profile that fell below the threshold (5). In the Fig. 2 example, there are holes in the mask for bins representing the ranges 0-9, 10-19, and 80-89. This anomaly mask became the mechanism used to determine if a future measurement was anomalous enough to be reported (see fourth level of Figure 2). For example, (8) and (10) represent measurements of future instances that are tested against the anomaly mask. Instance (8) has 90 packets, and when tested against the mask, the mask blocks this (9) from being reported. However, Instance (10) has 83 packets, and when tested against the mask falls through the hole (11) to become an anomalous event (12) that is reported to the NSM security officer (16).

136. NSM's measurements of recent activity were statistical descriptions of recent activity. I understand that SRI has claimed that a current snapshot of recent activity is not a short-term statistical profile. That is incorrect. In fact, Mr. Valdes, one of the named inventors, testified that a short-term profile could be a single measurement or event.[80]

137. In addition, under Symantec's alternative claim construction, the NSM would satisfy the requirement that the long-term statistical profile was an exponentially aged probability distribution of historical values. Although NSM's short-term statistical description was not aged, *NSM 1990* specifically pointed the reader to SRI's IDES

[79] '338 col. 6:59-7:3.
[80] Valdes Tr. 376-377.

project, which became the NIDES project. The NIDES project had extensive publications about the algorithms for such a short-term statistical profile, and one of skill would have been motivated to combine the two systems.[81]

138. In addition to detecting anomalous activity, NSM applied a set of rules, or signatures, to look for specific patterns of behavior. *NSM 1990* states that signatures are particularly important when the intrusion detection system is first deployed and has not had time to build up useful profiles. To implement the signatures, the NSM used the same masking technology developed for the anomaly detection, as shown in Fig. 2. Examples of rules mentioned in the paper include:

- Looking for connections with small numbers of packets indicating a possible failed login.
- Looking for a host connecting to large numbers of other hosts indicating a host probing the network.

139. The final analysis method mentioned in *NSM 1990* is another form of anomaly detection, but instead of measuring the behavior of an object (e.g., by measuring the number of packets associated with the object), the NSM measured the probability of the object just existing. This analysis was only applied to connections, and the measurement is the probability of seeing a connection from a host **A** to host **B** via service **C**. This triple (which is essentially the Source-Destination-Service class of objects) is called a "data path." When a new connection was observed, the NSM looked up a profile to see if this data path had been used in the past. If not, then the connection was considered anomalous. In *NSM 1990* the data path profile was relatively simple: a data path was considered normal if it had been used at least once in the previous two weeks. Subsequent implementations used an exponentially aged distribution of previously-observed connection patterns.

[81] *See, e.g.*, *Statistical Methods*.

140. When the NSM detected suspicious behavior (through the analysis of anomalous behavior, anomalous data path, or signature detection) it responded in two different ways. First, the NSM could raise an alarm for the security officer. Second, the NSM could alter the analysis of collected data. *NSM 1990* describes this as a "monte carlo divide-and-conquer search" of the data collected. The idea was motivated by the concern that the computer that the NSM was running on would not have the computational power to perform anomaly detection on all objects all the time, so it would only initially perform the anomaly detection at the higher-level groupings (e.g., Source or Source-Destination objects), and the NSM would drill down and examine the more detailed objects (e.g., Source-Destination-Service or Connection objects) only if one of the larger groupings appeared suspicious. Thus, if the NSM detected anomalous behavior at the Source-Destination object, it could alter its planned analysis to also include examining the Source-Destination-Service objects underneath it.

141. *NSM 1990* also described plans to extend the approach to "hybrid systems" that involved "host-based monitors to watch over the activities of individual hosts."[82] *NSM 1990* discussed the need to extend the approach of the current implementation to "distributed monitoring of wide area networks," where the "network monitoring functions have to be distributed among several nodes" and these nodes would "exchange information to reach a consensus on whether an attack is in progress."[83]

142. In my opinion, as more fully reflected in Exhibit C to my report, *NSM 1990* as a publication satisfied every limitation of the indicated claims and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the inventions claimed in the '338 patent. Consequently, the noted claims of the '338 patent were not novel when filed.

[82] *NSM 1990* at 302.
[83] *NSM 1990* at 302-303.

143. By 1991 NSM was in public use. At trial I may testify regarding the use of NSM on the UC Davis campus and at several government-owned locations. In my opinion, as more fully reflected in Exhibit C to my report, this public use of NSM anticipates the indicated asserted claims of the patents-in-suit.

144. In addition, as reflect in Exhibit C, all of the rest of the asserted claims of the '338 patent are invalid due to obviousness.

145. As described elsewhere in my report, it would have been obvious to one of ordinary skill in the art to combine NSM with other intrusion detection projects—including DIDS, GrIDS, and ISM—developed in whole or in part by researchers at UC Davis. In fact, NSM was part of both the DIDS and ISM architectures, and documentation associated with GrIDS explicitly describes using GrIDS in conjunction with NSM.[84] One of skill would have been motivated to combine these related systems, all of which were directed at detecting computer intrusions, to improve the functionality of the overall system.

146. It also would have been obvious to one of ordinary skill in the art to use the statistical profile-based anomaly detection of network traffic data practiced by NSM to analyze the additional network traffic data categories analyzed and described by: "A Network Security Monitor—Final Report," NetRanger, ISS RealSecure, Network Flight Recorder, Network Security Probe, Gabriel, synkill, SNMP/RMON IETF standards, and the SunScreen Firewall.[85] As shown in these references, all of these network traffic data categories were well-known, and it was also well-known that these categories could be analyzed using either signature detection, statistical profile-based anomaly detection, or both. One of skill would have been motivated to combine NSM with these additional

[84] *See* GrIDS webpage discussing using GrIDS with NSM. [SYM_P_0512095, 2096]
[85] *See* the NSM chart at Ex. C for the full titles and dates of all these references.

references in order to monitor additional types of network traffic for improved detection capabilities.

147. Furthermore, as previously discussed, it would also have been obvious to use the NIDES algorithms to implement the statistical profile-based anomaly detection practice by NSM. Indeed, *NSM 1990* explicitly suggests "looking at generating masks using the techniques used by IDES, Wisdom and Sense, and other intrusion detection systems, so that we can make an experimental analysis of these different methods."[86] One of skill would have been motivated to combine NSM with the NIDES algorithms based upon this explicit suggestion.

148. It would have been obvious to one of ordinary skill in the art to use the statistical profile-based anomaly detection of network traffic data practiced by NSM with distributed, hierarchical intrusion detection systems like DIDS, NetRanger, ISS RealSecure, CIDF, AIS, ISM, GrIDS, and Emerald 1997. *NSM 1990* explicitly suggests extending NSM to a distributed environment.[87] One of skill would have been motivated to combine NSM with these additional systems because both NSM and these other systems were designed for detecting computer intrusions, and NSM explicitly suggested such a combination.

149. In addition, to the extent there were any differences between the configuration of NSM and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of NSM based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity

[86] *NSM 1990* at 299.

[87] *See NSM 1990* at 302-03. Indeed, several designers of distributed intrusion detection systems either used or expressed an interest in using NSM within such a system. *See* discussion of DIDS, ISM and GrIDS. *See also Emerald 1997* at 364.

in very large enterprise networks. For example, as the NSM's primary developer, I went on to develop Network Radar, which did in fact address many of these differences.

2. DIDS

150. The Distributed Intrusion Detection System (DIDS) was a combined network and host-based intrusion detection system first developed in the early 1990s at UC Davis, LLNL, and Haystack Labs. Many different publications have described various aspects of the DIDS system, as listed in Exhibit D. This printed publication portion of this report focuses on DIDS as it was described in: Steven Snapp et al., "Intrusion Detection Systems (IDS): A Survey of Existing Systems and A Proposed Distributed IDS Architecture" (February 1991) ("*DIDS Feb. 1991*") and S.R. Snapp, J. Brentano, G.V. Dias, L.T. Heberlein, C. Ho, K.N. Levitt, B. Mukherjee, (with S.E. Smaha, T. Grance, D.M. Teal, D.L. Mansur), "DIDS -- Motivation, Architecture, and an Early Prototype" Proc. 14th National Computer Security Conference, Washington, DC, Oct. 1991, pp. 167-176 ("*DIDS Oct. 1991*").

151. DIDS was a two-level, heterogeneous intrusion detection system, as shown in Figure 3.[88] At the first level, traditional network-based and host intrusion detection systems detected suspicious activity for the subjects that they monitored. These monitors used combinations of methods such as misuse and anomaly detection to detect suspicious activity. At the second level, the DIDS Director correlated activity across the network using a construct called the Network Identifier, or NID. The Director used the NID to aggregate/integrate reports of suspicious activity from the multiple host and LAN monitors. The Director could respond to suspicious activity by alerting a security officer

[88] As the inventors themselves have stated "[f]urther research by UC Davis in the Distributed Intrusion Detection System (DIDS) [23] and later Graph-based Intrusion Detection System (GRIDS) [24] projects has attempted to extend intrusion monitoring capabilities beyond LAN analysis, to provide multi-LAN and very large-scale network coverage." *See* Live Traffic Analysis at p. 3.

or directing the individual host and LAN monitors to modify their analysis. Finally, the papers documented future plans for DIDS including monitors for network gateways and file servers, and additional levels of hierarchy.

152. The NSM was the network monitor being used in the DIDS system. *DIDS Oct. 1991* explained that NSM was the LAN monitor being used and explicitly mentioned profiles, and further directed the reader to *NSM 1990.*[89] *DIDS Feb. 1991* also explained that NSM was the LAN manager, and again directed the reader to *NSM 1990.*[90] Given this explicit direction to use *NSM 1990* to build the DIDS LAN monitor, *NSM 1990* is incorporated into the text of each DIDS publication. In the alternative, the combination of the two is obvious, as it is explicitly suggested by the references themselves. The actual DIDS implementation contained NSM and all of its anomaly detection capabilities.[91]

153. As documented in *DIDS Feb. 1991* and *DIDS Oct. 1991*, DIDS had the following salient features:

- Performed intrusion detection across multiple monitors (hence the term "Distributed" in "Distributed Intrusion Detection System").

- Performed two-level hierarchical monitoring, with LAN and host monitors at the first level and the Director at the second level.

- Correlated activity across multiple hosts and network connections via the Network Identifier (NID).

- Aggregated suspicious activity across multiple users, hosts, and network connections.

[89] *DIDS Oct. 1991* at 167, 175.
[90] *DIDS Feb. 1991* at 1, 15.
[91] *See* Terrance Lee Goan Jr., "Towards a Dynamic System for Accountability and Intrusion Detection in a Networked Environment," M.S. Thesis, Division of Computer Science, University of California, Davis, 1992 at 35.

- Coordinated multiple host monitors (based on Haystack) and network monitors (based on NSM).
- Detected suspicious activity on hosts and in network connections using signature and anomaly detection.
- Designed to support multiple network monitors and multiple host monitors spanning multiple LANs.

154. DIDS was designed to aggregate all potentially suspicious activity across a network performed by a single user to that one user, no matter how many different hosts, user names, and network connections were used by that individual. Previously, host-based intrusion detection systems largely analyzed actions by individual user names independently from other user names on the same system or even the same user name on different hosts. Similarly, network-based intrusion detection systems largely analyzed connections independently from other connections. For example, if there was a connection from host A to host B and another connection from host B to host C, a network-based intrusion detection system would analyze these connections independently.

155. DIDS LAN monitors (NSMs) monitored the same categories of network traffic discussed previously in numerous NSM publications. These categories were thus already known to one of skill reading the DIDS publications. There was no need in the DIDS publications to reinvent the wheel. Furthermore, as shown in Exhibit D, both DIDS references explicitly discuss monitoring multiple different categories of network traffic, such as network connections and network packet data volume.

156. Figure 4 demonstrates the problem DIDS addressed. A user (1) sits down at Host1 and logs in as user Bob (2). Eventually the user connects to Host2 (3) and logs in as Eve (4). The user also connects to Host3 (5) and logs in as user Bob (6). At some point the user executes a "substitute user" (su) command (7) to become user Mary (8) on

Host3. The user, from the Mary account on Host3, tries to connect to Host4 (9) and Host5 (10). The user, from the Eve account on Host2, also tries to connect to Host6 (11).

157. Prior to DIDS, a typical collection of host-based intrusion detection systems would report activity from four different users on three different machines (plus perhaps activity from hosts 4, 5, and 6). Meanwhile, a typical network-based intrusion detection system would report five different connections. DIDS first goal was to perform correlation to recognize that these nine activities originate from an initial login point. DIDS second goal was to aggregate all the suspicious activity from these activities to this initial login point.

158. DIDS performed correlation by having host and network-based sensors report low-level audit events associated with logins, user name changes, and network activity to a centralized sensor called the DIDS Director. The DIDS Director had a rule-based system that used each of these low-level audit events to construct a single Network Identifier (NID). The NID was the mechanism of correlation. DIDS performed aggregation/integration by having each of the host and network-based sensors report suspicious activity to the DIDS Director. Because the DIDS Director could map activity from each user on each machine and each network connection to a single NID, any and all suspicious activity reports associated with the same NID could be aggregated together. This allowed activity that might be only slightly suspicious at any point in the network to be aggregated together and become clearly suspicious. The correlation of activity across the network via the construction of the NID was a primary focus of all the DIDS papers.

159. As shown in Figure 5, DIDS had three types of monitors - host monitors, LAN monitors, and a Director - and each had its own data sources. The host monitor processed audit records generated by the host's operating system (1). Similarly, the LAN monitor processed packets read from the network (1). Both the host and LAN monitors generated two types of data processed by the Director: low-level events (2) and reports

of suspicious behaviors (3). The low-level events (2) consisted of events such as user logins from the host monitor and connection starts from the LAN monitor, and the Director used these low-level events to construct the NID. The host and LAN monitors also generated reports of suspicious behaviors for the subjects they measured (e.g., user activity for the host monitors and connections for the LAN monitors), and the Director aggregated these suspicious reports together to arrive at an aggregate suspicion measure for the NID.

160. The host and LAN monitors in DIDS used well-known methods for determining if the subjects they monitored were behaving suspiciously or not. These methods included misuse detection (e.g., using signatures) and anomaly detection (e.g., using profiles). The Director used a rule-based system to correlate the low-level events into the NID and looked for suspicious connectivity patterns.

161. The DIDS Director could respond to suspicious behavior by alerting a system security officer, who in turn could direct DIDS to actively respond to an attack including cutting off network access to a particular user. The DIDS Director could also direct the host and network monitors to modify their analysis via the "SET" command.

162. The NSM that formed the LAN monitor in DIDS was typically deployed at the gateway between the organization's internal hosts and the Internet. The largest deployment of network-based intrusion detection systems was by the Air Force Information Warfare Center (AFIWC) and NSM was deployed at an Air Force base's gateway between the base's internal networks and the Internet. This deployment in AFIWC was typically just in front of the firewall. Furthermore, *DIDS Oct. 1991* explicitly directs one to monitor at a gateway.

163. The DIDS papers described three levels of capability, each of which are represented in Figure 3:

- **DIDS Original Design** – DIDS, as described in *DIDS Oct. 1991*, was designed to support multiple LANs, each with its own set of LAN and host monitors. In addition to this being specifically mentioned in the text,[92] the LAN monitor reports to the Director included a field to identify *which* LAN monitor the report came from.

- **DIDS First Implementation** – While DIDS was designed to support multiple LANs, the first implementation to be delivered to the customer at the end of the first two-year contract was only tested on and guaranteed for a single LAN.

- **Planned Future DIDS** – *DIDS Oct. 1991* also described extending DIDS to include the addition of special-purpose monitors for network gateways and servers (i.e., file servers) and supporting additional levels of hierarchical monitoring.

164. Under Symantec's alternative claim construction of "monitor" the DIDS system is still anticipatory. The DIDS LAN monitor was software that could be reconfigured, for example, to add certain string matching capabilities. The DIDS Director was similarly software that could add such capabilities. The LAN monitors collected, analyzed and responded to suspicious network activity – using analysis engines and functionality to respond to the detection of events. The DIDS Director collected both suspicious events from the monitors, and low-level information (including network information) to track a user moving through a network. The DIDS Director correlated various suspicion reports and analyzed connectivity information for suspicious patterns. The DIDS Director GUI provided response by alerting the user of problems.

165. In my opinion, as more fully reflected in Exhibit D to my report, the DIDS system disclosed in *DIDS Oct. 1991 and DIDS Feb. 1991* satisfied every limitation of the indicated claims and enabled one of ordinary skill prior to November 9, 1997 to practice the inventions claimed in the patents-in-suit. Consequently, the noted claims of the patents-in-suit were not novel when filed.

[92] *DIDS Oct. 1991* at 169, *see also DIDS Feb. 1991* at 1 which refers to "LAN managers" indicating more than one network monitor, and 11 "each LAN segment has a LAN monitor…"

166. By 1991 DIDS was in public use. At trial I may testify regarding the use of DIDS on the UC Davis campus and at a government-owned location. In my opinion, as more fully reflected in Exhibit D to my Report, this public use of DIDS anticipates the noted claims of the patents-in-suit.

167. In addition, as reflect in Exhibit D, the remaining asserted claims of the patents-in-suit are invalid due to obviousness.

168. As described elsewhere in my report, it would have been obvious to one of ordinary skill in the art to combine DIDS with other intrusion detection projects—including NSM, GrIDS, and ISM—developed in whole or in part by researchers at UC Davis. Professor Karl Levitt, a highly-regarded researcher in intrusion detection, was the principal investigator for all of these projects and publicly disclosed these projects repeatedly in conferences and presentations. In fact, NSM served as the DIDS LAN Monitor, and DIDS was also part of the ISM architecture.[93] One of skill would have been motivated to combine these related systems, all of which were directed at detecting computer intrusions, to improve the functionality of the overall system.[94] See the NSM, ISM, and GrIDS charts for such references.

169. It would have also been obvious to one of ordinary skill in the art to use both the statistical profile-based anomaly detection of network traffic data practiced by DIDS and the signature-based detection of network traffic practiced by DIDS to analyze the network traffic data categories analyzed and described by: NetRanger, ISS RealSecure, Network Flight Recorder, Network Security Probe, Gabriel, synkill, and the SunScreen Firewall.[95] As shown in these references, all of these network traffic data categories were well-known, and it was also well-known that these categories could be

[93] *See ISM 1992.*
[94] *See* GrIDS webpage discussing using GrIDS with NSM. [SYM_P_0512095-2096].
[95] *See* the DIDS chart at Ex. D for the full titles and dates of all these references.

analyzed using either signatures detection, statistical profile-based anomaly detection, or both. One of skill would have been motivated to combine DIDS with these additional references in order to monitor additional types of network traffic for improved detection capabilities.

170. It would have been obvious to one of ordinary skill in the art to add to DIDS the response capabilities practiced or described by NetRanger, ISS RealSecure, CIDF, AIS, Network Security Probe, the '750 Patent, and synkill. Such response capabilities were well-known, and one of skill would have been motivated to combine DIDS with such response capabilities in order to expand the capabilities of DIDS to respond to intrusions.

171. It also would have been obvious to one of ordinary skill in the art to use DIDS to monitor a variety of different network entities, including virtual private network nodes and firewalls. Both *DIDS Feb. 1991* and *DIDS Oct. 1991* explicitly describe monitoring a variety of network nodes, and one of skill would have been motivated to expand the set of nodes monitored in order to increase the ability of DIDS to monitor all nodes in a network.

172. It would have also been obvious to one of ordinary skill in the art to extend the DIDS architecture to include peer-to-peer communications, as practiced by ISM, CSM, and Network Security Probe. Indeed, the DIDS was a part of the ISM architecture, and *ISM* 1992 described a peer-to-peer communications scheme. Furthermore, the CSM literature explicitly discusses DIDS.[96]

173. In addition, to the extent there were any differences between the configuration of DIDS and the configuration described in the patents-in-suit, it would

[96] *See* G. White et al., "*Cooperating Security Managers: A Peer-Based Intrusion Detection System*," IEEE Network, Jan./Feb. 1996.

have been obvious to modify the configuration of DIDS based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

174. Thus, as indicated in Ex. D, the remaining asserted claims of the patents-in-suit are obvious.

3. ISM

175. This report covers the Internet Security Monitor design, or ISM, as described in the paper "Internetwork Security Monitor: An Intrusion-Detection System for Large-Scale Networks," published in Proc. 15th National Computer Security Conference. Washington, DC, Oct. 1992, pp. 262-271 ("*ISM 1992*"). ISM had the following salient features:

- Built on NSM and DIDS capabilities and presumed familiarity with these systems (includes references).
- Correlated user's activity across domains through extension of DIDS' Network Identifier (NID).
- Developed the "thumbprint" technology to support NID construction with network-only monitors.
- Aggregated user's suspiciousness across domains.
- Aggregated host and service suspiciousness across domains.
- Supported peer-to-peer domain coordination.
- Supported hierarchical domain coordination.
- Balanced the value of aggregation with the need of privacy across domains.

176. UC Davis developed the ISM model as an extension to the Distributed Intrusion Detection System (DIDS). ISM's design was initially developed in 1992

towards the end of the first phase of DIDS, and it was conceived as a logical extension to the DIDS architecture. ISM first introduced the "thumbprint" concept (explained later in this section). UC Davis folded the NSM, DIDS and ISM concepts into their first DARPA-sponsored intrusion detection project, "Intrusion Detection for Very Large Networks," during which the thumbprint work was enhanced. A second paper on thumbprinting was published in 1995.[97] DARPA began funding the "Intrusion Detection for Very Large Networks" in approximately 1993. Eventually in the fall of 1995 the DARPA effort was redesigned into what became known as the Graph-based Intrusion Detection System (GrIDS), and work continued until about 1999.

177. *ISM 1992* explicitly states that ISM uses the DIDS and NSM systems and has an architecture "based on NSM and DIDs."[98] *ISM 1992* also points the reader to *DIDS Oct. 1991* and another paper on NSM: L.T. Heberlein, B. Mukherjee, K.N. Levitt, D. Mansur, "Towards Detecting Intrusions in a Networked Environment," Proc. 14th Department of Energy Computer Security Group Conference, pp. 17.47-17.65, May 1991 ("*NSM 1991*"). In addition, as noted previously, *DIDS Oct. 1991* explicitly directed the reader to *NSM 1990*. These references clearly state that NSM used both anomaly and misuse detection. Given this explicit direction to use *NSM 1990, NSM 1991* and *DIDS Oct. 1991*, these papers are incorporated into the text of *ISM 1992*. In the alternative, the combination of these references is obvious, as it is explicitly suggested by the references themselves.

178. DIDS was designed to detect and track attacks that spanned multiple users, multiple hosts, and multiple connections, but only within a single administrative domain. ISM was designed to detect and track attacks that spanned multiple domains. ISM

[97] *See* S. Staniford-Chen, and L.T. Heberlein *"Holding Intruders Accountable on the Internet"* Proc. of the 1995 IEEE Symposium on Security and Privacy, Oakland, CA, 8-10 May 1995, pp. 39-49.
[98] *ISM 1992* at 262, 263, 268.

presented extensions to the DIDS architecture to support the scalability and privacy concerns associated with coordinating multiple, independently administered domains.

179. *ISM 1992* introduced two extensions to support the ISM architecture: thumbprints for tracing users across unmonitored hosts and a set of protocols to extend "the Distributed Intrusion Detection System (DIDS) (see [Sna 91]) into arbitrarily wide networks."[99]

180. The thumbprint concept was designed to track users moving through any number of unmonitored hosts. In fact, two network connections can be compared even if there are an unknown number of intermediate hops separating the two connections being compared. For example, suppose an attacker begins at host H1, logs into host H2, from there logs into host H3, and so on until the attacker logs into host H10 from host H9. The thumbprint mechanism can compare connections H1->H2 and H9->H10 and determine they are related (i.e., it correlates them) even without any knowledge of all the connections from H2 to H9. Thus, the thumbprint approach extends DIDS' Network Identifiers (NIDs) across hosts without host monitors.

181. ISM's second extension to DIDS was a set of protocols to extend correlation and aggregation across multiple domains. The set of protocols could be divided into two groups: one to support peer-to-peer communications and another group to support hierarchical communications.

182. The peer-to-peer protocol extensions provided two capabilities: (1) to extend the tracking of a NID across multiple, independently administered domains and (2) to provide summaries of suspiciousness for NIDs, hosts, services, and vulnerabilities.

183. The hierarchical protocol extends the peer-to-peer communication by allowing hierarchically higher monitors to collect more information than peers can,

[99] *ISM 1992* at 264, citing to *DIDS Oct. 1991*.

namely, collecting additional details about a user's (NID) movements through the domain and lower-level events associated with that NID such as the number of files the user opened.

184. Figure 6 shows the basic approach. At the lowest level of the hierarchy, network monitors (1) and host monitors reported to a local ISM monitor such as the DIDS Director (2). These local ISM monitors could optionally report to ISM monitors in other administrative domains or to hierarchically higher ISM monitors (3). ISM monitors communicated with other ISM monitors in other administrative domains through peer-to-peer messaging (4) based on the CMIP protocol. ISM monitors communicated with hierarchically higher monitors (5) by extending the peer-to-peer messages with additional messages to collect details not available to peer monitors.

185. As the ISM had an "architecture based on NSM and DIDS," it used as data sources both network packets and host audit records; although with ISM's thumbprinting approach to track users, the ISM could support a network-only domain monitor.

186. At the lowest level of the ISM hierarchy, the host and network monitors responded by forwarding reports of suspicious reports to the local Director. For each subsequent level in the ISM hierarchy, or for peer-to-peer communications between ISM monitors, an ISM monitor responded by making the summary of their analyses available to other monitors.

187. *ISM 1992* presents an architecture that extends DIDS "into arbitrarily wide networks"[100] using both peer-to-peer and hierarchical communications. At the lowest level of the hierarchy, NSM-based monitors analyzed network traffic using a combination of anomaly and misuse detection. These network monitors reported to local ISM monitors. ISM monitors could communicate with other peer monitors in other

[100] *ISM 1992* at 264.

administrative domains or with hierarchically higher monitors. ISM monitors extended DIDS' tracking of uses via the NID concept across administrative domains, *and ISM 1992* proposed the concept of thumbprints to support this tracking across unmonitored hosts. ISM monitors could correlate and aggregate suspicious behavior based on NIDs, hosts, services, and vulnerabilities.

188. With regard to Symantec's alternative claim construction of "monitor," the prior analysis I provided on DIDS would apply to ISM as well. As noted in *ISM 1992*, at each level of the hierarchy there would be a security workbench. This security workbench would allow network managers to log in to their local ISM domain monitor and modify analyses. Furthermore, under SRI's definition of "statistical detection method" the ISM thumbprinting method would also qualify as a statistical detection method. However, under Symantec's claim construction for this term, thumbprinting would not qualify.

189. In my opinion, as more fully reflected in Exhibit E to my report, the ISM system disclosed in *ISM 1992* satisfied every limitation of the indicated claims and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the inventions claimed in the patents-in-suit. Consequently, the noted claims of the patents-in-suit were not novel when filed.

190. In addition, as reflected in Exhibit E, the remaining asserted claims of the patents-in-suit are invalid due to obviousness, for the same reasons as discussed previously regarding DIDS.

4. GrIDS

191. This section covers the Graph-based Intrusion Detection System (GrIDS system) and the prior art publications Staniford-Chen, S., et al. "GrIDS - A graph based intrusion detection system for large networks," 19th National Information Systems

Security Conference, 1996 ("GrIDS 1996") and "Steven Cheung, Rick Crawford, Mark Dilger, Jeremy Frank, Jim Hoagland, Karl Levitt, Stuart Staniford-Chen, Raymond Yip, Dan Zerkle, "The Design of GRIDS: A Graph-Based Intrusion Detection System," Technical report, UC Davis Department of Computer Science, Davis California (May 14, 1997) ("GrIDS 1997").

192. UC Davis conceived and developed the Graph-based Intrusion Detection System (GrIDS). UC Davis began development of GrIDS in 1995 as part of an initial 1993 DARPA contract to develop intrusion detection systems for very large networks. UC Davis continued the GrIDS work under a 1996 contract, and continued to work on it until about 1999. The GrIDS system was running publicly at UC Davis by approximately early 1997.

193. GrIDS had the following salient features:

- Correlated and aggregated intrusion detection reports via hierarchical collection of GrIDS monitors.
- Focused on detecting attacks that spanned multiple connections and across multiple administrative domains.
- Assumed the existence of lower-level sensors such as intrusion detection systems, but was not tied to any specific sensor type.
- Correlated reports from a single lower-level sensor (e.g., an IDS) or multiple lower-level sensors by building a graph of related reports.
- Aggregated reports from a single-lower level sensor (e.g., an IDS) or multiple lower-level sensors by summarizing attributes associated with the correlated reports.
- Reduced data amounts passed to hierarchically higher GrIDS monitors by only passing up relevant summarized information.
- Correlated reports from multiple lower-level GrIDS monitors arranged in a hierarchical fashion, typically organized to reflect an organization's hierarchical structure.

- Aggregated reports from multiple lower-level GrIDS monitors by integrating the summarized attributes from the lower-level GrIDS monitors into the graphs built by the higher-level GrIDS monitors.

- Detected intrusive activity by size and shape of graphs.

- Supported detecting intrusive activity by aggregating sensor-supplied attributes provided in individual reports.

- Configurable through built-in functions and operators to refine correlation and aggregation.

- Extensible through user-supplied functions to refine correlation and aggregation.

- Responded to suspicious behavior by sending messages to hierarchically higher monitors and possibly a user interface.

- Performed data reduction by sending only reduced graphs to hierarchically higher monitors.

- Described two network-based sensors for demonstrating GrIDS correlation and aggregation capability.

194. GrIDS was designed to detect and track attacks that spanned multiple domains. Whereas DIDS was designed to detect and track attacks that spanned multiple users, multiple hosts, and multiple connections, it only did so within a single administrative domain. Furthermore, the DIDS Director was tightly bound to the type of information generated by the lower-level sensor information. GrIDS, on the other hand, was largely designed to be independent of the lower-level sensors and could correlate and aggregate information from a wide variety of sources.

195. GrIDS was a hierarchical monitoring system for which the bottom-most layer of the hierarchy was made up of non-GrIDS sensors. These bottom-most sensors could be intrusion detection systems (for which there were a large number that had been developed and/or deployed by this point), simple network sniffers, or any other type of monitor that could send its output to a GrIDS monitor. GrIDS supported a well-defined and public protocol, including the "GrIDS Common Packet Format (GCPF)" and a well-

defined syntax, so third-party sensors could be extended to support GrIDS or the sensors could be "wrapped" with a second program that translated the sensor's proprietary message into a GrIDS message.

196. Once a GrIDS monitor received a report from a lower level sensor, it attempted to correlate the new report with previously received reports. If a correlation could be found, the new report was aggregated with the previous reports. GrIDS performed correlation and aggregation through the concept of a graph – a computer science term referring to a data structure consisting of nodes and edges connecting nodes. Each new report was turned into a graph; for example, a report of a network attack from host A to host B would be turned into a report with nodes representing hosts A and B and an edge between these nodes representing the communication between these nodes (the edge could be "annotated" with the specific type of attack). This new graph representing the report was compared to existing graphs representing past reports, and if the new graph shared a node or edge with an existing graph, the new graph was "correlated" with this existing graph, and the existing graph was extended by "aggregating" or "integrating" the new graph with it.

197. GrIDS monitors used a number of heuristic and potentially user-supplied functions to determine if a given graph was suspicious and should be forwarded to higher-level GrIDS monitors or user interfaces. An example heuristic was a graph that exceeded a particular size in the number of nodes or number of edges. For example, if the graph grew too large, it indicated a potential worm or attacker moving through the network, and that graph would be determined to be suspicious. If a graph took on a particular shape, that would indicate a sweep or other pattern of misuse, and again the graph would be flagged as suspicious.

198. If a GrIDS monitor determined that a graph had become suspicious, the GrIDS monitor automatically responded by forwarding a reduced version of the graph to

a hierarchically higher GrIDS monitor. It could also send an alert to a graphical user interface or any other subject that subscribed to receive alerts.

199. Hierarchically higher GrIDS monitors applied the same algorithms and code, but they applied them to the graphs passed up from hierarchically lower GrIDS monitors (as opposed to sensors such as traditional intrusion detection systems).

200. Figure 7, based on *GrIDS 1997* figures 1.3 and 1.4, demonstrates several features of GrIDS. Fig. 7 shows the College of Engineering at a university, and the College is composed of three departments – Electrical Engineering (EE), Computer Science (CS), and Civil Engineering (CE). Each department has its own GrIDS monitor receiving messages from one or more sensors. Each department GrIDS monitor sends alerts (consisting of reduced graphs) to the hierarchically higher college of engineering GrIDS monitor.

201. In Fig. 7, the hosts are labeled 'A' through 'J'. Host 'A' in Civil Engineering initiates a connection to host 'B' (1). Shortly after this activity, host 'B' initiates a connection to host 'H' in Electrical Engineering (2). Host 'H' then initiates connections to hosts 'J' (3) and 'I'. The sensor (4) in EE detects these last several connections and based upon its own analysis reports them as suspicious via standard GrIDS messages (5) to the department's GrIDS monitor (6). The EE GrIDS monitor builds the graph shown in (7). It determines this graph is suspicious and sends a GrIDS message (8) to the hierarchically higher college GrIDS monitor (9). The college monitor builds its own graph (10).

202. Fig. 7 demonstrates a 3-tier hierarchy consisting of the lowest level sensor (4), the department GrIDS monitor (6), and the college GrIDS monitor (9). It also demonstrates correlation and aggregation through the construction of graphs from simpler reports (7) and (10). It also demonstrates communication over well-defined and open

protocols (5) and (8). It also demonstrates data reduction as the hierarchically higher college GrIDS monitor builds graphs from reduced data graphs (10).

203. GrIDS monitors processed messages encoded in GrIDS Common Packet Format (GCPF), and the data messages were either generated by lower-level sensors or hierarchically lower GrIDS monitors. The initial messages could "come from other IDSs, network sniffers, or any monitor that is equipped with a filter to send its output to GrIDS."[101]

204. *GrIDS 1997* describes a prototype network sniffer to demonstrate the GrIDS approach.[102] The sniffer analyzed TCP connections and UDP sessions. For TCP connections and UDP sessions the sniffer generated event reports for:

- Start of TCP connection
- Normal close of connection via FIN flags
- Close of connection via RST flags
- Close of connection because of a timeout
- Start of UDP sessions
- End of UDP sessions
- ICMP error messages

205. Furthermore, for Telnet connections the sniffer reported several additional Telnet protocol specific events including:

- START
- OPTION NEGOTIATION
- AUTHENTICATION

[101] *GrIDS 1997* at 7.
[102] *GrIDS 1997* at Chapter 7.

- DATA
- END
- RESET

206. Finally, for NFS sessions the sniffer reported several NFS protocol specific events including:

- NFS Authentication error
- NFS Stale file handle error
- NFS mount success
- NFS mount failure
- Set UID error
- Read success
- Write success

207. The proposed sniffer analyzed network packets, identified and tracked individual sessions (connections), and detected and reported the above-mention events. Each GrIDS monitor mapped a report from a lower-level sensor or GrIDS monitor into a graph and correlated and aggregated the new graph with the previous graphs as described above. A GrIDS monitor could measure several features of a graph to determine if it was suspicious (e.g., graph size) or use user-supplied functions to analyze the graph.

208. With regard to Symantec's alternative claim construction of "network monitor," the GrIDS graph engine would satisfy all the requirements for such a monitor. GrIDS graph engines could be dynamically reconfigured: for example, they could add hosts, add departments, or even accept an entire new rulebase. GrIDS graph engines collected, analyzed and responded to suspicious network activity. GrIDS graph engines included an analysis engine and a mechanism for response. GrIDS graph engines would

in fact use the same code base for all of the engines at each level of the hierarchy, and thus would also satisfy ISS's claim construction requirement of "generic code."

209. In my opinion, as more fully reflected in Exhibit E to my report, the GrIDS system disclosed in *GrIDS 1996* and *GrIDS 1997* satisfied every limitation of the indicated claims and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the inventions claimed in the patents-in-suit. Consequently, the noted claims of the patents-in-suit were not novel when filed.

210. Furthermore, GrIDS—as well as NSM and DIDS—was in public use prior to November 9, 1997. In my opinion, as more fully reflected in Exhibit E, this public use anticipates the indicated claims from the patents-in-suit.

211. In addition, as reflect in Exhibit E, the remaining listed claims of the patents-in-suit are invalid due to obviousness. As described elsewhere in my report, it would have been obvious to one of ordinary skill in the art to combine GrIDS with other intrusion detection projects—including NSM, DIDS, and ISM—developed in whole or in part by researchers at UC Davis.

212. It also would have been obvious to one of ordinary skill in the art to use GrIDS to monitor a variety of different network entities—including routers, gateways, and firewalls—as practiced or described by DIDS, the SunScreen Firewall, and NetRanger. A critical function of the GrIDS sensor is to monitor traffic in and out of its domain, and the best place to do that is at the gateway.

213. In addition, to the extent there were any differences between the configuration of GrIDS and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of GrIDS based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a

desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

214. Thus, as indicated in Exhibit E, the indicated asserted claims of the patents-in-suit are obvious.

5. JiNao

215. This section covers the JiNao design as described in the technical report by F. Jou, "Architecture Design of a Scalable Intrusion Detection System for the Emerging Network Infrastructure," dated April 1997 ("*JiNao Report*"), as well as the associated slides indicated on Exhibit G. MCNC and North Carolina State University (NCSU) performed this work. The JiNao design provided a scalable architecture to detect and protect against both known and unknown attacks against the network infrastructure.

216. The architecture described, collectively called JiNao, used a combination of anomaly detection, misuse detection, correlation, and aggregation to detect intrusive activity. The lowest level JiNao monitors (described as a "Local JiNao") analyzed packets, while the higher level monitors (described as "Remote Management Applications") analyzed the results of Local JiNao monitors or other higher level monitors. Communications between monitors was via the standard Simple Network Management Protocol (SNMP) and Management Information Base (MIB) architecture. JiNao also uses the standard SNMP network protocol and MIB architecture to support hierarchical and peer-to-peer communications to provide scalability and detect larger scale attacks. The JiNao architecture was independent of any particular network protocol (the modules use the generic term "Protocol Data Unit" (PDU)), but the developers planned to deliver implementations for analyzing the OSPF routing protocol and the